University of Tripoli Faculty of engineering Electrical &electronic Eng. Dept

Course :EE334 FIRST EXAM Spring 2018

==========

Q-1) Identify the choice that best completes the statement or answers the question

1)What are the names of the 4 segment registers?

a) Data, Index, Code, Stack c) Stack, Extra, Code, Data

b) Stack, Data, Base, Counter d) Stack, Index, Extra, Code

2)Since the intel 86 processor has an address bus of 20 -bits, its memory is segmented into 1 M segments (i.e. 2^{20}).

a) True b) False

3)Which flag(s) does the 80x86 use to check for unsigned arithmetic overflow?

a) Overflow b) Direction c) Interrupt d) Carry

4) Which register is used as an offset address for the string instruction destination in the microprocessor?

a) DI b) SI c) BP d) DX

5) If CS = 7FA2H , SS = 0801H, SI = 0100H and IP = 438EH the address of the next instruction is:

a) 83DAEH b) 83DA0H c) 438EH d) None of the above

6) Assume that the AX register contains the value 6521 H. What will be the contents of AX after execution the instruction: SUB AL, AH

a) 65BCH b) 4421H c) BC21H d) 6544H

7) Which of the following is an illegal 8086 instruction?

a) add ax,[cx] b) mov ax,[bx] c) inc [si] d) aDd bx,[dx]

8) Let W be an array of Words, one of the following is a correct code to set the fifth element in W to 100.

a) mov [W+10],100 b) mov [W+2*4],100 c) mov [W+5],100 d) None of the above

9) What does the NOT instruction do?

a) Two's Complement b) One's Complement

10) What will be the contents of register AL after the following has been executed

MOV BX, F78Ch MOV AL, 7Eh ADD AL, BL

a) 6A and carry flag is set b) 6A and carry flag is reset

c) 0A and carry flag is set d) 0A and carry flag is reset

0.5 Each

b) What are 8086 16-bit registers and what are the specific features of the accumulator, index registers, instruction pointer , and flags register?

AX :: used for I/O operations and string manipulation plus arithmetic operation>

BX ::

CX ::

DX ::

SI :: used as source data addresses in string manipulation instructions

DI :: used as destination data addresses in string manipulation instructions

BP ::

SP ::

IP :: pointing to the next instruction to be fetched.

flag register : indicate the status of the microprocessor.

2.5

Q-2) The 8 data bytes are stored from memory location E000H to E007H. Write 8086 Assembly Language code to transfer the block of data to new location B001H to B008H using indirect addressing.

```
Mov SI,  E000h
Mov DI , B001h
Mov   AX, [SI]      ; FIRST WORD
MOV [DI] , AX

ADD SI , 2
```

4

```
ADD DI , 2
Mov   AX, [SI]      ; SECOND WORD
MOV [DI] , AX
ADD SI , 2
ADD DI , 2
Mov   AX, [SI]      ; THIRD WORD
MOV [DI] , AX
 ADD SI , 2
ADD DI , 2
 Mov   AX, [SI]      ; FOURTH WORD
MOV [DI] , AX
```

Q-3) Write assembly code to do the following :

a) Declare an array named Y of 10 signed double words initialized to 1, 2, ..., 10

Y DD 1,2,3,4,5,6,7,8,9,10

b) Declare a Null-terminated string named prompt_msg with a message "Enter YourPassword"

PROMPT_MSG DB "Enter YourPassword$"

c) Declare 4-bytes array named Marks of 100 numbers, all initialized to 0.

MARKS DB 100 DUP 0

2

d) Declare a string variable named SysInfo containing the word "TEST" repeated 500 times.

SYSINFO DW 500 DUP "TEST$"

Q-4) Write complete assembly program that computes the following equation

$$\sum_{i=1}^{n} n * i + 1\%(\frac{n}{i})$$

4.5

Q-5) Write a complete assembly program that calculates the sum of array signed numbers larger than a given value (sample).In your code: Use relative addressing memory mode.

Assume that the number of items in the array is equal to 12345H

7

```
data segment
    array      DD 00000000,
               3086 dup (BBBBBBBBh, CCCCCCCh, 11111111h,AAAAAAAAh)
    givenvalue DD XXXXXXXXh
    sum        DD 00000000H
    arraySize  DD 12345H
data  ends
code segment
    assume Cs:code ,DS:data
start:
   mov SI ,offset array
   mov DI ,offset givenvalue
   mov BX ,offset sum
   mov cx ,0FFFFh
```

```
  x :
    mov  ax    ,[si+0]
    mov  BP    ,[si+2]
    mov  dx    ,BP
    CMP  ax    ,[DI+0]
    SBB  BP    ,[DI+2]
    JNG  Y
        add  [bx]  ,ax
        adc  [bx+2],dx
        Y:
        add  si    ,4
        loop x
  mov cx ,2344h
  xx :
    mov  ax    ,[si+0]
    mov  BP    ,[si+2]
    mov  dx    ,BP
    CMP  ax    ,[DI+0]
    SBB  BP    ,[DI+2]
    JNG  YY
          add  [bx]  ,ax
          adc  [bx+2],dx
        YY:
          add  si   ,4
   loop xx
code ends
end start
```

**********GOOOOOD LUCK **********